HRUSHIKESH VASISTA
Chicago, IL 60616
+1 (312) 478-8173 | hmvasista@gmail.com | https://www.linkedin.com/in/hvasista

SUMMARY

- **4.5** years of experience in software design and development
- 3.5 years in AUTOSAR COMStack-BSW and ASW development, 1 year in HMI modeling
- Experience in requirement analysis, unit testing, supporting multiple projects
- Communication protocols: **CAN**
- Programming | Scripting: **C, C++, Java, Android | OAW, Perl**
- Tools: **Eclipse, Git, Android Studio, EBGuide, EBTresos, CANoe, RADAR, IBM DOORS, ClearQuest, EASEE-BASD**
- International languages: **German** (A2 certified)

EDUCATION

Aug 2015 - Present *Graduate Student,* **Illinois Institute of Technology**, Chicago, IL
Master of Science, Computer Science
GPA 3

Coursework Introduction to Advanced Studies(data-structures in Java), Introduction to Algorithms, Computer Networks, Mobile Application Development (Android)

On-campus work experience
Student Assistant *(Jun 2016 - Aug 2016)*
AMGP Coordinator *(Aug 2016 - Aug 2016)*

Aug 2007 - Jun 2011 **Visvesvaraya Technological University**, Belagavi, India
Bachelor of Engineering, Electronics and Communications Engineering
Score 78.33%

PROFESSIONAL EXPERIENCE

Jun 2014 - Dec 2015 *Staff Software Engineer,* **Visteon Technical Services Centre,** Pune, India

KEY TASKS

- **Implementation for Autosar based Application SW for HUD**
  HUD (Heads-up Display) is a Driver Information feature. I developed ASWs for features like Adaptive Cruise Control, alerts and warnings, navigation etc

- **HMI Modeling for HUD**
  The HMI (Human-Machine Interface) is about what is to be displayed on HUD to the user; this is done based on the input from user / internal events triggered.

- Performed testing of integrated project for HMI and Combiner operations alongside Integrator
- Supported HMI modeling for multiple HUD projects

OTHER TASKS

- Developed a Perl based script for reliable merging of multiple CAN .dbc files

Jun 2011 - May 2014 *Software Engineer,* **Robert Bosch Engg. and Business Solutions**, Coimbatore, India

KEY TASKS

- **Implementation of Post-Build Selectable feature for CanIf module**
  With post-build it is possible to have a single .hex file which can be flashed on the required ECU and decide which configuration set shall be active at installation time. This was implemented as a vendor specific feature in AR4.0 with all other modules remaining Pre-compile.

- **FD-CAN support in CanIf**
  FD-CAN is the 'next version' of CAN which supports switching to a higher baudrate during payload and CRC. This feature was implemented as a vendor specific feature in AR4.0

- **Design and implementation of CanSM module as per AR4.0**
  The CanSM had significant changes in its state machines in AR4.0 over the earlier AR3.1 specification - inclusion of Prepare states for NO_COMM and FULL_COMM, state machine for baudrate change, managing PN states/ flags of transceivers, mandatory mode indications from controller and transceiver to change states and minor changes in Busoff recovery state machine, with corresponding changes in CanIf.

- **Optimization of CanSM 4.0 state machine implementation**
  The CanSM 4.0 state machine follows two paths based on the mode indication received from the device; immediately after a request (synchronous) and with a delay (asynchronous). State machine to proceed such that both possibilities are handled and satisfies the timing as per AR3.1 in the best-case scenario. Also aimed at improving code readability and maintainability.

- **Partial Networking (PN) support in CanSM and CanIf modules as per AR4.0**
  Partial Networking is a state in a CAN system where some nodes are in low power mode while other nodes are communicating. This reduces the power consumption by the entire network. Nodes in the lowpower modes are woken up by pre-defined wakeup frames. PN is a configurable feature in CanIf and CanSM which is provided by CAN transceivers.

- **Implementation of scalable test cases (simulation) for CanSM module**
  Scalable test cases ease the testing for all the possible CanSM network states at the same time and thus improve code coverage. A new set of functions were written and maintained as a library, thus reusable. Simplifies the process of testing in future (addition or modification of test cases)

OTHER TASKS

- A Perl based tool to analyze the log entries from *.asc into *.xls format simplifying the analysis of huge CANoe logs (1-2 GB)
- A Perl based tool to automate the build-test sequence
- Provided training on AUTOSAR Com-Stack and other support to the new hires in the team
- Reviewer for initial COMASSO releases

ACADEMIC PROJECTS

May 2016 — **Illinois Institute of Technology**, **Chicago, IL**

GoShop - a grocery delivery App for Android phones

- Helped team setup the project through Git
- Integrator for the project
- Responsible for implementing sign-in (Google and Facebook), user registration, cart pages
- Implemented the multi-language support (prototype: English, German)
- https://github.com/HrushikeshVasista/GoShop

Nov 2015 — **Illinois Institute of Technology**, **Chicago, IL**

Simulation of Link-state routing algorithm

- Implemented the 'modified' version of Dijkstra's algorithm - to find **all** shortest possible paths between 2 nodes - in Java
- Performed unit and system testing for the project

OTHER ACTIVITIES

May 2016 - current — **Hobby project**

Full stack Web development

Intended to improve knowledge on back-end and front-end development

Jan 2013 - Apr 2014 — **Hobby project**

matrix.h

C++ APIs to realize basic matrix operations (https://github.com/HrushikeshVasista)